Neural Magic Release Notes 1.1 and 1.0

1 message

Neural Magic <communications@neuralmagic.com>
Reply-to: info@neuralmagic.com
To: maxieds@gmail.com

Thu, Sep 1, 2022 at 1:03 PM

Hi there!

Neural Magic has been busy this summer working on the Community Edition (CE) of our DeepSparse Engine and SparseML libraries; we're excited to share highlights of releases 1.0 and 1.1.

Our 1.0 release was a huge milestone and we could not have gotten here without all of your support and feedback! The full technical release notes are always available within our GitHub release indexes linked from the specific Neural Magic repository.

If you have any questions, need assistance, or simply want to say hello to our vibrant ML performance community, join us in the Deep Sparse Community Slack. We are growing the community member by member and are happy to see you there.

1.1 Release

Sparse Use Cases: Information Retrieval (New), Zero-Shot Text Classification (New), Named-Entity-Recognition (Improved)

We've uploaded models to the SparseZoo and created new pipelines for information retrieval (Haystack), zero-shot text classification (with transformers), and named entity recognition (with transformers). We developed a YOLACT pipeline for DeepSparse deployments. We added a CustomTaskPipeline to enable easier custom pipeline creation for our users.

DeepSparse pipelines now support dynamic batch, dynamic shape through bucketing, and asynchronous execution. Lastly, we also introduced inference performance for:

- Unstructured sparse-quantized transformer models
- Slow activation functions (such as Gelu or Swish) when they follow a QuantizeLinear operator
- Some sparse 1D convolutions. Speedups of up to 3x are observed
- Squeeze, when operating on a single axis

1.0 Release

CLI support for transformers, YOLO5, and Torchvision deployments + Docker updates!

Through our standardized CLI formats, we've added new support for transformers, YOLOv5, and Torchvision. This allows users to get started with these models through well-documented CLI flows.

For more effective docker management and deployments, we've added new Dockerfiles and Docker build processes into this release.

Additionally in the 1.0 release:

- Pruning mask creator deployed for use in PyTorch pruning modifiers
- Masked_language_modeling training CLI was added for transformers.
- Documentation additions made across all standard integrations and pathways
- Revamped testing to aid ease of contribution

Review the full release notes here.

Next Up

ARM support is on the way!

We are excited to announce that our engineering team is underway with development to build ARM support into the DeepSparse Engine, targeting AWS Graviton first. Initial results are very promising. We are seeking Alpha testers to test our initial implementations of the DeepSparse Engine running on ARM. If you are interested in joining our Alpha testing program, please respond to this email or contact Rob Greenberg via our community Slack.

Simplifying Model Optimizations with Sparsify

We want to make sparsity the new standard in ML models, but creating sparse, performant models is hard. Between understanding pruning and quantization techniques, evaluating tradeoffs in hyperparameter tuning, and finally fitting a model to your data, sparse model creation can be a headache.

We are going to change that.

We are working on a new product called Sparsify which provides pathways to grab or create sparse models that meet your business needs.

Have a generic NLP sentiment analysis task or an object detection problem with a class contained in the COCO dataset? Get started with the sparsify.package API to deliver a sparse model ready to tackle your ML task right out of the gate.

Do you already have a complete dataset and a business case that you are trying to solve, but just can't get a model that meets your accuracy needs or performance

requirements? Try out the sparsify.auto API where you just need to feed in your dataset, use case, and optional target optimization metrics, retrain on your infrastructure, and the sparsify.auto API will generate a sparse-transfer learned model for your use case on your dataset.

If this flow sounds interesting to you, we are looking for early testers! Please reply to this email or reach out to Rob Greenberg via our community Slack to get started with the alpha version of the Sparsify API in the next few weeks.

Until next time,

Neural Magic's Product Team

Neural Magic, 55 Davis Sq, Somerville, MA 02144, United States